

BAB 3

METODE PENELITIAN

Penelitian ini menggunakan metode Naive Bayes Classifier untuk menganalisis sentimen investasi obligasi *FR* pada *Twitter*. Penelitian ini membutuhkan data *tweet* yang diperoleh dari *Twitter* yang terkait dengan obligasi *FR*, dan kemudian mengolah data tersebut melalui proses preprocessing untuk mencapai hasil yang diinginkan. Data tersebut nantinya dapat memetakan sentimen netizen terkait investasi obligasi *FR*. Data *tweet* dikumpulkan dari *Twitter*, kemudian dilakukan preprocessing. Berikut bahan, alat, dan jalan penelitian yang terstruktur untuk menganalisis sentimen menggunakan data *tweet*.

3.1 BAHAN PENELITIAN

Bahan penelitian adalah data *tweet* dan *re-tweet* di *Twitter* terkait dengan investasi obligasi *FR*. Data *tweet* yang diambil selama periode 1 November 2016 – 11 Juni 2023 dengan jumlah data sebanyak 466 data *tweet*.

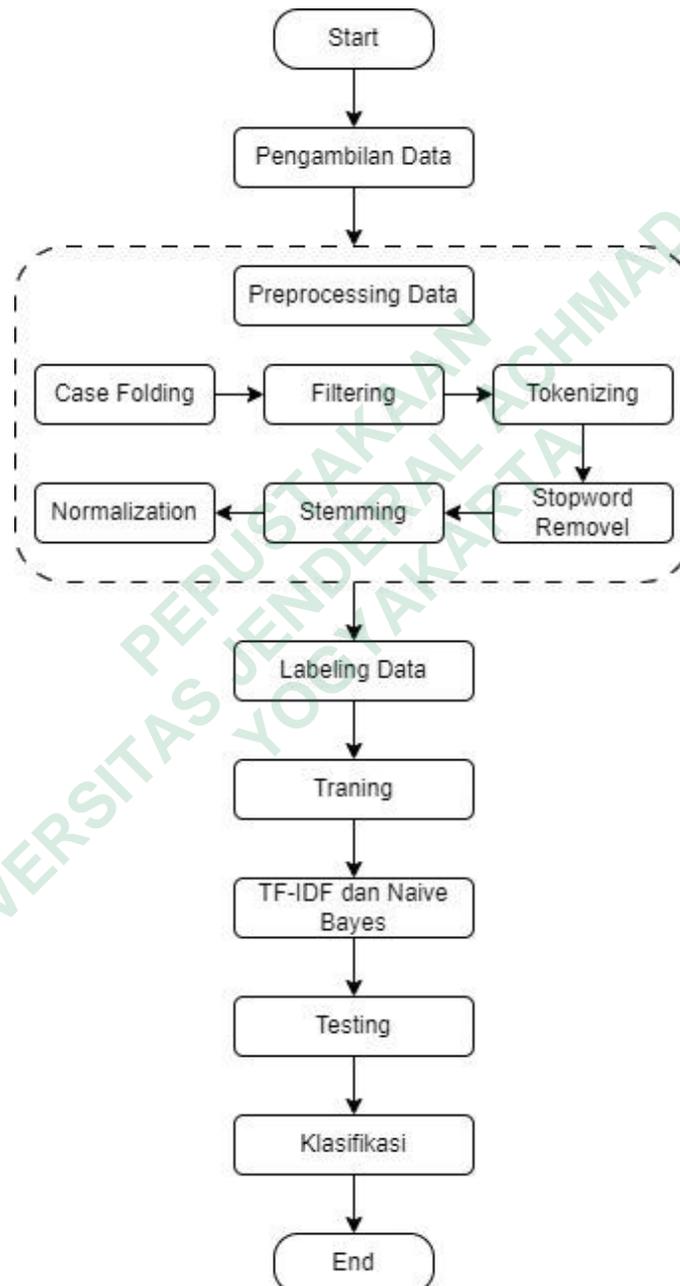
3.2 ALAT PENELITIAN

Alat yang digunakan dalam penelitian ini meliputi laptop sebagai perangkat hardware untuk mencari data dan menjalankan sistem informasi yang digunakan dalam penelitian. serta dukungan jaringan internet sebagai sarana berselancar dan mengambil data dari media sosial di internet. Berikut Sistem Operasi dan software yang dipergunakan dalam penelitian ini adalah :

1. Sistem Operasi: Windows 10
2. Google colab
3. Website apify
4. Anaconda versi 3
5. Jupyter Notebook
6. Microsoft Excel
7. *Twitter*

3.3 JALAN PENELITIAN

Penelitian ini menggunakan bahasa pemrograman python, Anaconda 3 dan Jupyter Notebook kemudian data ditampilkan menggunakan Microsoft Officer Excel dan dimodelkan dengan bantuan *library* pada bahasa pemrograman python. Alur diagram penelitian ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

3.3.1 Pengambilan data

Proses pengambilan data menggunakan *library* *snsrape* dan website *apify.com*. Penggunaan *library* *snsrape* dan *apify* digunakan untuk mengumpulkan data dari media sosial *twitter*. *Library* *snsrape* ini hanya bisa dijalankan menggunakan *google colab* dengan menginstall *library* *snsrape* pada bahasa pemrograman *twitter*. Kemudian website *apify* adalah sebuah platform yang menyediakan layanan web scraping, data extraction, dan automation serta menyediakan fitur dan *library* untuk pengambilan data pada *twitter*, fitur yang digunakan dalam penelitian ini yaitu *twitter* scraper. Contoh data *tweet* yang sudah diambil dapat dilihat pada tabel 3.1.

Tabel 3.1 Contoh Data *Tweet*

No	Text
1	@bibitid coba jelasin obligasi <i>FR</i> dong. Gak ngerti ngerti aku min
2	Langkah pertama, menentukan pembanding Ada ratusan obligasi dan reksa dana pendapatan tetap di Yg dipilih adalah <i>FR</i> (Fixed Rate) Gov Bond dan seri syariahnya atau disebut <i>PBS</i> (Project Based Sukuk) Dan <i>RD</i> Pendapatan Tetap @reksadana_panin tempat saya mencari nafkah
4	nunda nikah sampe bangun kosan sendiri dulu, naroh duit di obligasi <i>fr</i> buat pendidikan anak, istri tenang, anak pun terjamin
5	Belum terlambat untuk siapin kebutuhan finansial jangka panjang lo, yuk mulai journey investasi Obligasi <i>FR</i> di @bibit.id sekarang!
6	<i>ST010</i> , Obligasi <i>FR</i> , dan Deposito, Mana yang Return-nya Lebih Tinggi? blog.bibit.id/blog-1/st010-o...
7	<i>Sbn</i> sama obligasi <i>fr</i> ni masih bingung ane . Apa milihnya hanya berdasarkan yield aja, atau kupon nya juga, dll.
9	Ada yg tau ga yaa bedanya <i>sbn</i> sama obligasi <i>fr</i> apa?? Udh baca" tp masih blm "klik"
10	Ada yg nggak tau itu obligasi. Pas ditanya wartawan, gelagapan

3.3.2 Preprocessing

Preprocessing data adalah pengolahan data teks yang ada dengan mengambil langkah untuk memperbaiki data teks yang tidak rapi dan masih belum ada perbaikan. Pada tahap preprocessing ini ada beberapa *library* yang membantu proses preprocessing, seperti *library* *pandas* untuk memanipulasi data, *library* *nlTK*

membantu dalam pemrosesan natural bahasa, dan *library* sastrawi untuk mereduksi infleksi kata dalam bahasa Indonesia. Sebelum dilakukan preprocessing data maka perlu dilakukan import *library* yang dapat dilihat pada Gambar Proses dalam preprocessing meliputi tahap :

1. Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil atau huruf besar. Kode dapat dilihat pada gambar 3.2.

```
def case_folding(text):
    lower_word = text.lower()
    return lower_word
dt_tweet['case_folding'] = dt_tweet['text'].str.lower()

print(dt_tweet['case_folding'])
```

Gambar 3.2 Program Case Folding

2. Filtering adalah proses menghapus karakter-karakter yang tidak diperlukan seperti tanda baca dan karakter khusus lainnya, tahap ini dapat dijabarkan sebagai berikut :

Special Removal adalah sebuah proses pembersihan teks pada *tweet* dengan cara menghapus karakter-karakter khusus atau simbol-simbol tertentu yang tidak relevan atau tidak diperlukan dalam Gambar 4 1 analisis sentimen. Dapat dilihat pada gambar 3.3.

```
def remove_tweet_special(case_folding):
    text = case_folding.replace('\t', " ").replace('\n', " ").replace('\ ', "")
    text = text.encode('ascii', 'replace').decode('ascii')
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)(\w+:\w+:\w+)", " ", text).split())
    return text.replace("http://", " ").replace("https://", " ")
dt_tweet['clean'] = dt_tweet['case_folding'].apply(remove_tweet_special)
```

Gambar 3.3 Program Special Removal

- 1) Number removal dilakukan untuk menghapus angka yang terdapat dalam teks data. Dapat dilihat pada gambar 3.4.

```
def remove_number(clean):
    text = re.sub(r"\d+", "", clean)
    return text
dt_tweet['clean'] = dt_tweet['clean'].apply(remove_number)
```

Gambar 3.4 Program Number Removal

- 2) Punctuation removal adalah penghapusan karakter khusus yang tidak memiliki pengaruh signifikan dalam analisis teks. Dapat dilihat pada gambar 3.5.

```
def remove_punctuation(clean):
    clean_spcl = re.compile('[/(){}\[ \]|@,;]')
    clean_symbol = re.compile('[^0-9a-z]')
    text = clean_spcl.sub('', clean)
    text = clean_symbol.sub(' ', clean)
    return text
dt_tweet['clean'] = dt_tweet['clean'].apply(remove_punctuation)
```

Gambar 3.5 Program Punctuation Removal

- 3) Whitespaces removal dilakukan untuk menghilangkan spasi yang berada di awal dan di akhir kalimat. Dapat dilihat pada gambar 3.6.

```
def remove_whitespace(clean):
    corrected = str(clean)
    corrected = re.sub(r"//t", r"\t", corrected)
    corrected = re.sub(r"(\s)+", r"\1", corrected)
    corrected = re.sub(r"(\n)+", r"\1", corrected)
    corrected = re.sub(r"(\r)+", r"\1", corrected)
    corrected = re.sub(r"(\t)+", r"\1", corrected)
    return corrected.strip(" ")
dt_tweet['clean'] = dt_tweet['clean'].apply(remove_whitespace)
```

Gambar 3.6 Program Whitespaces Removal

- 4) Single char removal adalah penghapusan karakter tunggal atau karakter yang hanya terdiri dari satu huruf dalam teks data. Kode dapat dilihat pada gambar 3.7.

```
def remove_singl_char(clean):
    singl = re.sub(r"\b[a-zA-Z]\b", "", clean)
    return singl
dt_tweet['filtering'] = dt_tweet['clean'].apply(remove_singl_char)
```

Gambar 3.7 Program Single Char

3. Tokenizing adalah proses memecah teks menjadi token atau kata-kata yang nantinya akan mempermudah melakukan Stopword Removal. Dapat dilihat pada gambar 3.8.

```
from nltk.tokenize import word_tokenize

def tokenizing(filtering):
    tokens = word_tokenize(filtering)
    return tokens

dt_tweet['token'] = dt_tweet['filtering'].apply(tokenizing)
print(dt_tweet['token'])
```

Gambar 3.8 Program Tokenizing

4. Stopword removal adalah proses menghapus kata-kata yang umum dan tidak memiliki makna seperti “dan”, “atau”, “yang”, dan sebagainya. Dapat dilihat pada gambar 3.9 .

```

from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

def stopword_removal(tokens):
    if isinstance(tokens, str):
        stopwords = stopword.remove(tokens)
    else:
        stopwords = [stopword.remove(token) for token in tokens]
    return stopwords
dt_tweet['stopword'] = dt_tweet['token'].apply(stopword_removal)

print(dt_tweet['stopword'])

```

Gambar 3.9 Program Stopword Removal

5. Stemming adalah proses mengubah kata-kata menjadi kata dasarnya. misalkan kata “menarik” setelah dilakukan stemming akan menghasilkan kata “tarik”. Dapat dilihat pada gambar 3.10 .

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

def stemming(stopwords):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(stopwords)
dt_tweet['stemmer'] = dt_tweet['stopword'].apply(lambda x: [stemming(token) for token in x])

print(dt_tweet['stemmer'])

```

Gambar 3.10 Program Stemming

6. Normalisasi adalah proses mengubah kata kata yang tidak baku menjadi kata-kata yang memiliki makna dan untuk menghasilkan representasi teks yang lebih seragam dan dapat diproses dengan lebih efektif dalam tahap analisis teks selanjutnya. Dapat dilihat pada gambar 3.11 .

```

def normalizing(tokens):
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in tokens]
    normalized_tokens = [norm_dict.get(token, token) for token in lemmatized_tokens]
    normalized_text = " ".join(normalized_tokens)
    return normalized_text

dt_tweet['normalizing'] = dt_tweet['stemmer'].apply(normalizing)

```

Gambar 3.11 Program Normalisasi

3.3.3 Labeling

Data *tweet* yang telah dipreprocessing akan dilakukan pelabelan dengan memberikan nilai sentiment positif atau negatif. Berdasarkan hasil pelabelan manual pada data *tweet*, diperoleh data *training* sebanyak 336 *tweet* yang sudah dilabeli dengan rincian 168 data *tweet* positif dan 168 data *tweet* negatif. Hasil pelabelan ditunjukkan pada tabel 3.2.

Tabel 3.2 Data Pelabelan

No	Clean_Data	Kelas	Label
1	saya dukung investasi obligasi fr saya tidak masalah kalau pasar platform tentu transparan sampai mungkin baik jelas lebih lengkap sesuai utama investor jual jatuh tempo	POSITIF	1
2	coba yang obligasi fr saya sih menaruh situ kalau bibit lama aman aja	POSITIF	1
3	mindahin tabung obligasi biasa obligasi fr fixed rate moga kuat terus bulan sampe jatuh tempo tahun tentu	POSITIF	1
4	market down harga obligasi fr turun mau buru beli biar mendapat yield lumayan eh apps welma nya down gimana nih	POSITIF	1
5	harga lebih kompetitif bibit mulai rp juta lebih rendah platform aman catat ksei beli guna rdn monitor obligasi fr real time jadi investor obligasi fr pertama nikmat fitur bibit plus	POSITIF	1
6	pimpin orientasi uang tidak heran kalau menteri manfaat umat khusus umat islam yang notabane besar indonesia mulai dari dana haji bangun jalan tol dengan beli obligasi seri fr yang aset adl jalan tol hingga beli obligasi bumh karya	NEGATIF	-1
7	ra dong bancak kasus korupsi inikasus hutang cek amp ricek sejak obligasi fr bkn fucking report lho amp sukuk pb jaman ngachiro yang terbit sejak okt rata jatuh tempo atas amp akhir yang atas	NEGATIF	-1
8	fr knp rezim kayak yang melarat ngutang plng gede blm obligasi dana haji embat dana bpjs embat duit calon umroh first travel raib pajak naik banyak terus keukeuh mau pindah ibu kota dg dana ratus trilyun rakyat sj hidup susah	NEGATIF	-1

9	tidak usah bicara soal obligasi daerah tuh obligasi perintah seri fr yang keluar yang yang underlying nya infrastruktur jalan tol amp obligasi bumh karya masuk jasa marga jsmr underlying jalan tol yang bangun bukti byk jalan tol jual yang bangun	NEGATIF	-1
10	kalau surat utang yang promosi influencer seri obligasi ritel yang minimal beli cmn rp jt amp kelipat nah kalau surat utang yang seri fr amp spn biasa beli investor besar amp institusi min beli rp	NEGATIF	-1

3.3.4 Training

Proses *training* data pada metode Naive Bayes Classifier dimulai dengan melakukan fitur ekstraksi pada data teks menggunakan TF-IDF untuk menghitung bobot pada setiap kata dan memudahkan proses Naïve Bayes dalam melakukan prediksi pada setiap kata. Setelah itu, dilakukan proses *training* data untuk membuat model klasifikasi yang dapat digunakan untuk melakukan klasifikasi sentimen secara otomatis.

1. TF-IDF

Perhitungan TF-IDF ini dilakukan menggunakan *library* Sklearn dan `TfidfVectorizer` pada Python. Dengan menggunakan fitur ekstraksi `TfidfVectorizer`, fungsi `feature_extraction` dibuat untuk mengubah dokumen menjadi data matriks fitur TF-IDF dan kemudian menghitung bobot keseluruhan dari kata-kata dalam dokumen. Data yang diambil kemudian disesuaikan dan dikonversi menggunakan fungsi `'fit.transform'` untuk mendapatkan data matriks hasil.

2. Naïve Bayes Classification

Bobot kata yang dihitung menggunakan TF-IDF digunakan untuk menghitung nilai probabilitas. *Library* Sklearn digunakan untuk mengimpor modul `MultinomialNB` yang secara otomatis melakukan perhitungan Naïve Bayes. `MultinomialNB` digunakan untuk menghitung nilai probabilitas setiap kelas dalam sebuah dokumen dengan memprediksi jumlah kemunculan kata-kata yang bersifat positif atau negatif secara otomatis. Setelah itu, dilakukan prediksi pada *tweet* yang masuk ke dalam kategori sentimen positif dengan menggunakan `predict_proba`. Setelah melakukan perhitungan TF-IDF.

3. Model klasifikasi

langkah selanjutnya adalah membuat model klasifikasi menggunakan variabel X dan y dari data pelatihan yang telah disediakan. Model ini dibuat sebagai fungsi agar lebih mudah dipanggil dan dieksekusi pada tahap-tahap berikutnya. Selanjutnya, model akan disimpan dalam bentuk file pickle agar dapat dibuka kembali dan digunakan di masa mendatang.

3.3.5 Testing

Testing adalah tahapan yang dilakukan untuk mengevaluasi tingkat akurasi pemodelan yang telah dibangun pada tahap *training* yang digunakan untuk memprediksi label atau kelas dari data uji. Pada tahap *testing*, model yang telah dibangun akan diuji dengan menggunakan data uji sebanyak 130 data *testing* yang belum diberi label nilai positif dan negatif. Confusion matrix pada penelitian digunakan untuk menghitung dan mengetahui presentase setiap pengujian yang dilakukan pada model yang sudah didapatkan. Menggunakan *library* sklearn untuk melakukan import confusion_matrix untuk melakukan *testing* dan prediksi. Metode yang digunakan antara lain :

- 1) accuracy digunakan untuk menghitung jumlah klasifikasi yang benar dibagi dengan total sampel yang diuji.
- 2) Precision digunakan untuk menghitung klasifikasi kategori positif yang benar dibagi dengan total sampel klasifikasi positif.
- 3) Recall digunakan untuk menghitung sampel yang diklasifikasikan kategori positif dibagi total sampel dalam *testing* yang berkategori positif.
- 4) f1-score digunakan untuk menghitung rata-rata dari precision dan recall